



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Christophe MANON

GAU:

SERIAL NO: 09/770,215

EXAMINER:

FILED: January 29, 2001

FOR: METHOD AND UNIT FOR CONTROLLING THE FLOW OF A TCP CONNECTION ON A FLOW CONTROLLED NETWORK

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS  
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number [US App No], filed [US App Dt], is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number , filed , is claimed pursuant to the provisions of 35 U.S.C. §119(e).
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
FRANCE	00 01735	February 11, 2000

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .  
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and  
(B) Application Serial No.(s)
  - ☐ are submitted herewith
  - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,  
MAIER & NEUSTADT, P.C.

\_\_\_\_\_  
Marvin J. Spivak

Registration No. 24,913

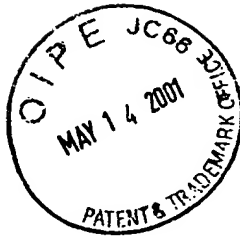
Joseph A. Scafetta, Jr.  
Registration No. 26,803



22850

**THIS PAGE BLANK (USPTO)**

#4



# BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 17 JAN 2001

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

Martine PLANCHE

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

**THIS PAGE BLANK (USPTO)**

Cet imprimé est à remplir lisiblement à l'encre noire

08 540 W / 260899

<b>REMISE DES PIÈCES</b> DATE <b>11 FEV 2000</b> LIEU <b>75 INPI PARIS</b> N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI <b>0001735</b> DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI <b>11 FEV. 2000</b>		<b>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE</b> À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE  <b>CABINET PLASSERAUD</b>  <b>84, rue d'Amsterdam</b> <b>75440 PARIS CEDEX 09</b>	
<b>Vos références pour ce dossier</b> (facultatif) <b>BFF000017</b>			
<b>Confirmation d'un dépôt par télécopie</b>		<input type="checkbox"/> N° attribué par l'INPI à la télécopie	
<b>2 NATURE DE LA DEMANDE</b>		<b>Cochez l'une des 4 cases suivantes</b>	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
<i>Demande de brevet initiale</i> <i>ou demande de certificat d'utilité initiale</i>		N° _____ Date   / / N° _____ Date   / /	
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>		<input type="checkbox"/> N° _____ Date   / /	
<b>3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)</b>  <b>PROCEDE ET UNITE DE CONTROLE DE FLUX D'UNE CONNEXION TCP SUR UN RESEAU A DEBIT CONTROLE</b>			
<b>4 DÉCLARATION DE PRIORITÉ</b> <b>OU REQUÊTE DU BÉNÉFICE DE</b> <b>LA DATE DE DÉPÔT D'UNE</b> <b>DEMANDE ANTÉRIEURE FRANÇAISE</b>		Pays ou organisation _____ N° _____ Date / / Pays ou organisation _____ N° _____ Date / / Pays ou organisation _____ N° _____ Date / / <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»	
<b>5 DEMANDEUR</b>		<input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»	
Nom ou dénomination sociale		MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTRE EUROPE B.V.	
Prénoms			
Forme juridique			
N° SIREN			
Code APE-NAF			
Adresse	Rue	Keienbergweg 58 1101 AG AMSTERDAM ZUIDOOST THE NETHERLANDS	
	Code postal et ville		
Pays		FRANCE PAYS-BAS	
Nationalité		Française NÉERLANDAISE	
N° de téléphone (facultatif)			
N° de télécopie (facultatif)			
Adresse électronique (facultatif)			

**BREVET D'INVENTION  
CERTIFICAT D'UTILITÉ**

REQUÊTE EN DÉLIVRANCE 2/2

REMISE DES PIÈCES DATE		Réserve à l'INPI	
LIEU		11 FEV 2000 75 INPI PARIS	
N° D'ENREGISTREMENT		0661735	
NATIONAL ATTRIBUÉ PAR L'INPI			
Vos références pour ce dossier : (facultatif)			
<b>6 MANDATAIRE</b>		BFF000017	
Nom			
Prénom			
Cabinet ou Société		Cabinet PLASSERAUD	
N° de pouvoir permanent et/ou de lien contractuel			
Adresse	Rue	84, rue d'Amsterdam	
	Code postal et ville	75009 PARIS	
N° de téléphone (facultatif)			
N° de télécopie (facultatif)			
Adresse électronique (facultatif)			
<b>7 INVENTEUR (S)</b>			
Les inventeurs sont les demandeurs		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	
<b>8 RAPPORT DE RECHERCHE</b>		Dans ce cas fournir une désignation d'inventeur(s) séparée	
Établissement immédiat ou établissement différé		Uniquement pour une demande de brevet (y compris division et transformation) <input checked="" type="checkbox"/> <input type="checkbox"/>	
Paiement échelonné de la redevance		Paiement en trois versements, uniquement pour les personnes physiques <input type="checkbox"/> Oui <input type="checkbox"/> Non	
<b>9 RÉDUCTION DU TAUX DES REDEVANCES</b>		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Requête antérieurement à ce dépôt (joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence)	
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes			
<b>10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE</b> (Nom et qualité du signataire) Stéphane VERDURE 97-0901		VISA DE LA PRÉFECTURE OU DE L'INPI	

DÉPARTEMENT DES BREVETS

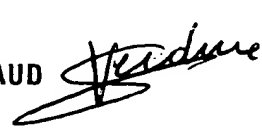
26 bis, rue de Saint Pétersbourg  
75800 Paris Cedex 08  
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° ... / ...

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

08 113 W / 260899

<b>Vos références pour ce dossier</b> (facultatif)		BFF000017	
<b>N° D'ENREGISTREMENT NATIONAL</b>		0001735	
<b>TITRE DE L'INVENTION</b> (200 caractères ou espaces maximum)			
PROCÉDE ET UNITÉ DE CONTRÔLE DE FLUX D'UNE CONNEXION TCP SUR UN RÉSEAU À DÉBIT CONTRÔLÉ			
<b>LE(S) DEMANDEUR(S) :</b>			
MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTRE EUROPE B.V.			
<b>DESIGNE(NT) EN TANT QU'INVENTEUR(S) :</b> (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
Nom		MANGIN Christophe	
Prénoms			
Adresse	Rue	17, rue du Calvaire 35310 BREAL SOUS MONFORT FRANCE	
	Code postal et ville		
Société d'appartenance (facultatif)			
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance (facultatif)			
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance (facultatif)			
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance (facultatif)			
<b>DATE ET SIGNATURE(S)</b> <b>DU (DES) DEMANDEUR(S)</b> <b>OU DU MANDATAIRE</b> (Nom et qualité du signataire)		Le 10 février 2000  CABINET PLASSERAUD   Stéphane VERDURE  97-0901	

**DOCUMENT COMPORTANT DES MODIFICATIONS**

[illegible]

Un changement apporté à la rédaction des revendications d'origine, sauf si celui-ci découle des dispositions de l'article R.612-36 du code de la Propriété Intellectuelle, est signalé par la mention « R.M. » (revendications modifiées).



## PROCEDE ET UNITE DE CONTROLE DE FLUX D'UNE CONNEXION TCP SUR UN RESEAU A DEBIT CONTROLE

La présente invention concerne un procédé et une unité de contrôle de flux d'une connexion TCP, en particulier une connexion TCP établie à travers un réseau à débit contrôlé.

Le protocole TCP (de l'anglais "Transmission Control Protocol"), ci-après  
5 TCP, est un protocole de la couche transport du modèle de référence TCP/IP. Il est très utilisé dans les réseaux et, notamment, le plus utilisé dans le réseau Internet. Il fournit un service de transport aux programmes d'application de la couche supérieure sous la forme de connexions bidirectionnelles transportant un flux d'octets entre un émetteur et un récepteur sur lesquels ces programmes  
10 sont exécutés (on parle aussi de transport de bout en bout). L'émetteur et le récepteur peuvent par exemple être respectivement un serveur de réseau et un terminal client. Parmi les protocoles reposant sur TCP, on peut citer : FTP (de l'anglais "File Transfer Protocol") pour les transferts de fichiers, HTTP (de l'anglais "HyperText Transfer Protocol") pour les transactions sur le réseau  
15 internet ou encore SMTP (de l'anglais "Simple Mail Protocol") pour le courrier électronique.

Pour assurer le transfert de données, TCP prévoit l'établissement préalable d'une connexion entre l'émetteur et le récepteur, dite connexion TCP, et la mise en œuvre de mécanismes de détection et de correction d'erreurs,  
20 ainsi qu'un mécanisme de contrôle de flux et un mécanisme de contrôle de congestion. C'est pourquoi on dit que TCP est un protocole orienté connexion (ou en mode connecté) qui, de plus, est fiable.

La connexion consiste à associer de façon bidirectionnelle un couple (adresse IP, port TCP) de l'émetteur à un couple (adresse IP, port TCP) du  
25 récepteur. Les adresse IP sont fournies par la couche IP alors que les ports TCP sont locaux. TCP découpe le flot d'octets à transmettre en segments dont la taille maximale est préalablement négociée entre les deux extrémités de la connexion TCP à l'ouverture de celle-ci. Un segment est donc l'unité de protocole de TCP. Des segments sont échangés pour l'établissement de la  
30 connexion (segment portant le fanion SYN), pour le transfert de données, pour leur acquittement (segment portant le fanion ACK), pour la fermeture de la

connexion (segment portant le fanion FIN) et pour la réinitialisation de connexion (segment portant le fanion RST).

Les segments transitant sur le sens descendant de la connexion (de l'émetteur vers le récepteur), ou segment de données, contiennent un numéro  
5 de séquence. Ce numéro de séquence est inséré par l'émetteur dans l'en-tête du segment de données. Il indique le rang, dans le flot de données transmis suivant le sens descendant de la connexion TCP, du premier élément de données (en pratique un octet) porté par le segment. Les segments de données sont acquittés positivement par des segments d'acquiescement (ou  
10 acquiescements) transitant sur le sens montant de la connexion (du récepteur vers l'émetteur). L'en-tête de chaque segment d'acquiescement contient un numéro de séquence, inséré par le récepteur. Ce numéro de séquence indique le rang, dans le flot de données transmis suivant le sens descendant de la connexion TCP, de l'élément de données ou octet suivant le dernier élément  
15 de données ou octet correctement reçu par le récepteur, c'est à dire le rang du prochain élément de données ou octet attendu par celui-ci.

Le mécanisme de contrôle de flux permet un contrôle du flux de bout en bout, c'est à dire entre les deux extrémités de la connexion, sans hypothèse sur la valeur du flux entre les différents nœuds de multiplexage du réseau. Afin  
20 d'améliorer l'efficacité de la transmission du flots de données, ce mécanisme est basé sur un système de fenêtre glissante. Le contrôle de congestion, quant à lui, agit sur la taille de cette fenêtre en fonction d'informations provenant du contrôle de flux qui permettent d'estimer le niveau de congestion du réseau. Une des caractéristiques principales du mécanisme de contrôle de flux de TCP  
25 est qu'il n'utilise aucune information explicite sur l'état du réseau fournie par ce dernier. A fortiori, le mécanisme de contrôle de congestion détermine l'état de congestion du réseau en fonction de signaux implicites fournis par le contrôle de flux, tels que l'arrivée d'acquiescements, l'expiration d'une temporisation ou encore la réception d'acquiescements dupliqués.

30 Les implications, sur les performances de TCP, des mécanismes de contrôle de flux et/ou de congestion dans le cadre, notamment, de réseaux de datagrammes purs, tels que IP (de l'anglais "Internet Protocol"), et de réseaux ATM (de l'anglais "Asynchronous Transfer Mode") font l'objet de nombreuses

recherches qui ont déjà abouti à des propositions de modification de ces mécanismes. Ces recherches ont principalement pour but d'améliorer l'utilisation du réseau et de permettre un partage équitable des ressources entre les différentes connexions TCP. L'invention se rapporte au domaine de ces travaux de recherche et concerne plus spécifiquement les techniques de

5 contrôle de flux des connexions TCP.

Le contrôle de flux de TCP repose sur un mécanisme de fenêtre glissante. La fenêtre  $W$ , exprimée en nombre d'octets, est le volume d'information que l'émetteur a transmis sans en avoir reçu l'acquittement par le

10 récepteur. Le principe de la fenêtre permet à TCP d'anticiper la réception d'un segment par le récepteur, c'est à dire d'envoyer des segments entrant dans la fenêtre sans en attendre l'acquittement. C'est pourquoi on parle de fenêtre d'anticipation ou de fenêtre de transmission. La taille de la fenêtre  $W$  est modulée par le contrôle de congestion et reste inférieure à une valeur

15 maximale fixée par le récepteur et annoncée dans chaque segment d'acquittement. A cet effet, chaque segment d'acquittement contient un paramètre de taille de fenêtre  $W_a$  dont la valeur correspond à ladite valeur maximale. Le fonctionnement de ce mécanisme est illustré par le schéma de la figure 1. Des segments de données sont référencés Seg.1 à Seg.11. Le sens

20 de transfert des segments de données depuis l'émetteur vers le récepteur est indiqué par une flèche. On notera que le mécanisme de fenêtre glissante de TCP opère sur des octets et non sur des segments, mais le segment étant en pratique l'unité de protocole de TCP, il est commode de se référer ici à des volumes de données exprimés en nombre de segments. La valeur maximale

25 de la taille de la fenêtre  $W_a$  annoncée par le récepteur dans un acquittement détermine une fenêtre offerte, qui, en combinaison avec la valeur du numéro de séquence présent dans le segment d'acquittement, couvre par exemple les segments Seg.4 à Seg.9. Le numéro de séquence présent dans l'acquittement indique dans l'exemple que le récepteur a déjà reçu et acquitté les segments

30 allant jusqu'au segment Seg.3 inclus. La valeur du paramètre  $W_a$  donne une indication de l'espace disponible dans une mémoire tampon d'entrée du récepteur, puisqu'elle correspond au volume de données que le récepteur déclare être prêt à accepter. L'émetteur calcule la fenêtre utilisable  $W_u$  définie

par  $W_u = W_a - W$ , et qui représente le nombre d'octets supplémentaires qu'il peut transmettre sans attendre l'acquittement des segments Seg.4 à Seg.6 déjà émis mais non encore acquittés. Dans l'exemple, cette fenêtre utilisable  $W_u$  couvre les segments Seg.7 à Seg.9. Ces segments peuvent donc être émis  
5 sans délai par l'émetteur.

Si l'on considère un réseau sans mémoire tampon et uniquement constitué de canaux de transmission caractérisés par un débit et un temps de transfert, la seule capacité mémoire offerte par le réseau pour absorber la fenêtre d'une connexion TCP est alors la mémoire équivalente du canal de  
10 transmission. Pour une connexion TCP déterminée, la taille de la fenêtre  $W$  correspond à la quantité de données envoyées dans le réseau et non encore acquittées. Physiquement, les données contenues dans le réseau à un instant déterminé comprennent d'une part les segments en cours de transfert sur le sens descendant de la connexion et d'autre part les acquittements en cours de  
15 transfert sur le sens montant de la connexion et concernant des segments déjà traités par le récepteur. La figure 2 illustre un exemple dans lequel le canal peut absorber une fenêtre correspondant à 12 segments. Dans cet exemple se trouvent dans le réseau à un instant déterminé, d'une part des segments Seg.7 à Seg.12 en cours de transfert sur le sens descendant (symbolisé par la flèche simple), et d'autre part des acquittements Ack.1 à Ack.6 en cours de transfert  
20 sur le sens montant (symbolisé par la flèche double). La réception d'un segment provoquant l'émission d'un acquittement par le récepteur, ces derniers sont alors espacés sur le sens montant d'une distance équivalente à la durée de transmission d'un segment sur le sens descendant.

25 Si le débit et le temps de transfert sont égaux sur les deux sens, la taille de la fenêtre  $W$  qui devrait au maximum être atteinte pour une connexion TCP est donnée par la relation suivante :

$$W_{\max} = d \times \text{RTT} \quad (1)$$

où  $W_{\max}$  est la taille maximale de la fenêtre (exprimé en octets),  $d$  est le  
30 débit de la connexion (exprimé en octets par seconde), et RTT (de l'anglais "Round-Trip Time") désigne le temps de transfert aller-retour des données sur le canal de transmission ou temps de boucle (exprimé en secondes). Le produit  $d \times \text{RTT}$  est encore appelé produit bande passante-temps de transfert. La

relation (1) fait apparaître le lien entre les performances de TCP en termes de débit de la connexion et la taille de la fenêtre maximale utilisable par cette connexion.

Le contrôle de congestion de TCP repose exclusivement sur des  
5 mécanismes de détection de perte de segment et de modulation de la taille de la fenêtre W en conséquence.

Les mécanismes de détection de perte de segment reposent d'une part sur l'expiration d'une temporisation de retransmission et d'autre part sur une détection anticipée. A chaque émission d'un segment, une temporisation de  
10 retransmission (RTO, de l'anglais "Retransmission Time-Out") est démarrée côté émetteur. Si la temporisation expire avant que l'acquittement correspondant n'ait été reçu, TCP considère que le segment est perdu et procède à sa retransmission en démarrant une nouvelle temporisation de retransmission plus élevée que la précédente. De plus TCP peut aussi  
15 supposer de manière anticipée qu'un segment est perdu, c'est à dire avant l'expiration de la temporisation de retransmission associée audit segment. Cette détection anticipée de la perte d'un segment tire parti du fait qu'un acquittement est systématiquement généré pour chaque segment reçu et qu'il porte le numéro de séquence du prochain segment attendu (le suivant dans  
20 l'ordre des segments). Suivant ce principe, un récepteur peut être amené à émettre des acquittements portant des numéros de séquence identiques (correspondant au dernier segment reçu dans l'ordre des segments) lorsque qu'une série de segments arrive dans le désordre. De tels acquittements (appelés acquittements dupliqués) permettent d'informer l'émetteur de la bonne  
25 réception des segments, à l'exception d'un ou plusieurs d'entre eux, reçus en contradiction avec leur ordre d'émission ou pas reçus du tout, et de l'informer du numéro de séquence du segment effectivement attendu. Du point de vue de l'émetteur, les acquittements dupliqués peuvent être attribués à différents problèmes du réseau : le fait que le réseau a réordonné les segments, qu'il  
30 a répliqué des acquittements, ou enfin qu'un segment a été perdu. Dans ce dernier cas, tous les segments suivant le segment supposé perdu génèrent un acquittement dupliqué.

Pour la modulation de la taille de sa fenêtre, un émetteur TCP utilise quatre algorithmes différents. Ces algorithmes sont appelés : démarrage lent ("slow start" en anglais), évitement de congestion ("congestion avoidance" en anglais), retransmission rapide ("fast retransmit" en anglais) et recouvrement rapide ("fast recovery", en anglais). Les deux premiers sont utilisés, soit immédiatement après l'établissement d'une connexion, lors du démarrage du transfert des données, soit à la suite de la perte d'un segment détectée par l'expiration de sa temporisation de retransmission. L'algorithme "slow start" a pour principal effet de forcer une réduction de la taille de la fenêtre  $W$  à un seul segment et de la faire croître ensuite d'un segment par acquittement reçu jusqu'à un seuil (appelé "slow start threshold" en anglais, ou "sssthresh") à partir duquel elle ne croît plus que d'une fraction de segment (égale à la taille d'un segment divisée par la taille courante de la fenêtre) par acquittement reçu. Cette deuxième phase d'accroissement a lieu selon l'algorithme "congestion avoidance" jusqu'au moment où la fenêtre  $W$  atteint la limite annoncée par le récepteur (taille de fenêtre  $W_a$  annoncée dans chaque acquittement). Typiquement, le seuil "sssthresh" est fixé à la taille maximale de fenêtre au début d'un transfert et à la moitié du volume de données émises et non encore acquittées suite à une perte de segment détectée par l'expiration de la temporisation de retransmission. Les algorithmes "fast retransmit" et "fast recovery", quant à eux, sont mis en œuvre lors de la détection anticipée d'une perte de segment. Dit autrement, ils sont déclenchés préventivement lors de la réception d'acquittement dupliqués pouvant laisser supposer la perte d'un ou plusieurs segments. Ils ont pour effet de réduire la taille de la fenêtre à la moitié du volume des données émises et non encore acquittées.

Bien que très robustes, les mécanismes de contrôle de flux et de contrôle de congestion de TCP peuvent entraîner une dégradation des performances du protocole. On constate ainsi diverses inconvénients. Premièrement, une connexion qui émet plus de données que les autres dispose d'une fenêtre plus grande que les autres. A l'inverse, une connexion qui utilise des segments de petite taille ou qui est soumise à des temps de transfert plus longs occupe moins vite les ressources disponibles dans le réseau. Il en résulte une répartition inéquitable, entre les différentes

connexions TCP, des ressources en débit et en mémoire du réseau. Deuxièmement, le temps de transfert de bout en bout est mal maîtrisé. En effet, aucune ressource du réseau n'étant préalablement allouée à la connexion et TCP n'exploitant aucune information explicite sur l'état du réseau, des points de congestion peuvent se déclarer, caractérisés par des temps de traversée de file d'attente non maîtrisés. En outre, le mécanisme de détection de perte par l'expiration de la temporisation de retransmission ainsi que la retransmission des segments perdus ont aussi un impact non maîtrisé en termes de temps de transfert. Troisièmement, une éventuelle congestion n'est détectée que tardivement, en raison précisément du caractère implicite de cette détection, reposant sur la détection de la perte d'un segment. Il s'ensuit qu'il n'est pas possible d'éviter la perte de segments et donc leur retransmission. Quatrièmement, et cet inconvénient découle du précédent, le débit utile est mal utilisé puisque les ressources de débit et de mémoire sont utilisées non seulement pour transporter les segments perdus jusqu'au point de congestion, mais aussi pour les retransmettre. Ce phénomène est encore accentué par la stratégie de retransmission dite "go-back-N" utilisée en cas de perte détectée par l'expiration de la temporisation de retransmission, qui consiste à retransmettre tous les segments à partir du segment perdu). Enfin, et cinquièmement, le flux de données est très irrégulier. Typiquement, tant qu'il n'a pas atteint sa taille de fenêtre maximale, TCP occupe de plus en plus de ressources de débit et de mémoire jusqu'à leur saturation. La perte de segments qui en découle donne lieu à des cycles de sous-utilisation du réseau (résultant des réductions brutales de taille de fenêtre dues aux algorithmes "slow start" et "fast recovery"), suivi d'une nouvelle phase d'accroissement de la taille des fenêtres à laquelle succède à nouveau un régime de perte, etc... D'un point de vue qualitatif, ces phénomènes se traduisent par une attente plus ou moins supportable lors du téléchargement de fichiers (utilisation par FTP), par le blocage des navigateurs lors de la consultation de sites sur internet (utilisation par HTTP), etc...

Plusieurs voies ont été explorées pour améliorer le comportement et les performances de TCP aussi bien au niveau de la couche transport (mécanisme d'extrémités) que de la couche réseau. Parmi celles-ci, les algorithmes FFR

(de l'anglais "fast retransmit/recovery") désignant les algorithmes décrits plus haut), l'utilisation d'acquittements sélectifs (SACK, de l'anglais "Selective ACKnowledgement") et la notification explicite de congestion via un bit d'indication de congestion ajouté aux paquets TCP (bit ECN, de l'anglais "Explicit Congestion Notification"), interviennent au niveau transport. FRR et SACK ont pour but d'améliorer le débit, d'éviter l'expiration des temporisations de retransmission et de favoriser une récupération rapide des ressources suite à une perte de segment. La technique ECN, quant à elle, requiert un support de la couche réseau pour agir sur la couche transport (indication explicite de congestion fournie par le réseau) et vise à améliorer l'équité de la répartition des ressources entre les connexions et à raccourcir les temps de transfert. Des mécanismes de couche réseau tels que les politiques de destruction anticipée de paquets (RED, de l'anglais "Random Early Discard"), la mise en œuvre d'échéanciers d'émission (WFQ, etc, ...) et les techniques de contrôle explicite du débit des connexions TCP permettent une amélioration des performances en termes de débit et d'équité de la répartition des ressources du réseau.

Le mécanisme de contrôle de la taille de la fenêtre qui est à la base du procédé selon l'invention peut se rattacher à cette dernière catégorie, à savoir celle des techniques de contrôle explicite du débit des connexions TCP.

Le débit d'une connexion TCP peut être contrôlé de plusieurs manières : soit par régulation du rythme d'émission des segments sur le sens descendant, soit par régulation du rythme de transmission des acquittements sur le sens montant. En effet, une fois la taille de sa fenêtre stabilisée, un émetteur TCP reçoit normalement un acquittement par segment émis, et il est autorisé à émettre sans délai un nouveau segment de données pour chaque segment d'acquittement reçu (régénération des crédits). Dit autrement, la première solution consiste à agir sur le rythme de consommation des crédits d'émission par l'émetteur et la seconde consiste à agir sur leur rythme de régénération. La seconde solution consiste à augmenter ou réduire l'espacement des acquittements et met en œuvre une fonction d'espacement des acquittements (appelée "Ack bucket"). La première solution peut consister, pour obtenir la conformité à un débit déterminé, à augmenter ou réduire l'espacement des émissions de segments. Mais elle peut aussi reposer sur



l'utilisation d'une couche de niveau inférieur fournissant des liaisons dont le débit est définissable et contrôlable. Tel est le cas, par exemple avec un réseau ATM ou TDM. Un point de contrôle du débit se trouve par exemple à l'interface du conduit ATM, au sein de l'équipement d'interconnexion avec le réseau local auquel est relié l'émetteur (équipement d'interconnexion source).  
5 Afin de compenser les différences de débit entre le réseau local et le réseau ATM, l'équipement d'interconnexion source comprend une mémoire pour contenir une file d'attente, réalisée sous forme d'une mémoire tampon. Les segments de données transmis par l'émetteur sur le sens descendant de la  
10 connexion TCP transitent par cette mémoire tampon.

Le procédé de contrôle de flux selon l'invention consiste en un mécanisme de contrôle de la taille de la fenêtre de transmission d'une connexion TCP, et s'applique plus particulièrement à une connexion TCP établie à travers un réseau à débit contrôlé. Au moins un point de contrôle du  
15 débit coopère avec une file d'attente par laquelle transitent les segments de données transmis par l'émetteur sur le sens descendant de la connexion. Un tel réseau est par exemple un réseau ATM à débit contrôlé avec une capacité de transfert de type CBR (de l'anglais "Constant Bit Rate"), ABR (de l'anglais "Available Bit Rate"), ou ABT (de l'anglais "ATM Block Transfer"). Un conduit  
20 ATM de ce type permet, une fois la ressource correspondante allouée dans le réseau, de limiter la taille des mémoires tampon mises en œuvre dans chaque étage de multiplexage du réseau et de réduire la perte sur le conduit à un niveau quasiment nul. Des connexions TCP établie à travers un tel conduit ATM, ne rencontrant aucune perte, font alors croître leur fenêtre de  
25 transmission W jusqu'à la taille maximale W<sub>a</sub> annoncée par le récepteur. Or, la capacité mémoire offerte par le conduit est très réduite (en supposant un débit et un temps de propagation raisonnables), et dépend du produit bande passante-temps de transfert. En effet, le volume V de données absorbable par le réseau, lorsqu'il est émis à un débit déterminé D sur un conduit caractérisé  
30 par un temps de transfert aller-retour T, est donné par la relation suivante :

$$V = D \times T \quad (2)$$

Toujours par souci de clarté, le volume V est dans la suite exprimé en nombre de segments, alors qu'il s'exprime en réalité en nombre d'octets, le

débit  $D$  s'exprimant en octets/seconde et le temps  $T$  en secondes. Le paramètre  $T$  étant un paramètre intrinsèque du réseau, il est considéré comme constant pour une connexion donnée. Par conséquent, si  $D$  est choisi tel que  $V$  est inférieur la taille de la fenêtre  $W$  de la connexion TCP, et si l'on considère  
5 que le cœur du réseau ne contient aucune mémoire tampon, on rencontre les situations suivantes :

- Dans le cas où les segments sont espacés, l'excédent de segments correspondant à la différence  $W-V$  s'accumule dans la file d'attente située au niveau du point de contrôle de débit,
- 10 • Dans le cas où les acquittements sont espacés, une file d'acquittements correspondant aux segments constituant l'excédent  $W-V$  s'établit au niveau du point de contrôle.

Les figures 3 et 4, sur lesquelles les mêmes éléments qu'à la figure 2 portent les mêmes références, illustrent respectivement ces deux situations  
15 dans le cadre d'un exemple dans lequel  $W=11$  segments et  $V=8$  segments. A la figure 3, la mémoire équivalente du réseau contient les segments Seg.5 à Seg.8 en cours de transfert sur le sens descendant et les acquittements Ack.1 à Ack.4 en cours de transfert sur le sens montant. Les segments de données excédentaires, à savoir les segments Seg.9 à Seg.11, forment une file  $F_s$   
20 stockée dans une mémoire tampon, en amont du point Pcd de contrôle de débit suivant le sens descendant. A la figure 4, la mémoire équivalente du réseau contient les segments Seg.8 à Seg.11 en cours de transfert sur le sens descendant et les acquittements Ack.4 à Ack.7 en cours de transfert sur le sens montant. Les acquittements excédentaires, à savoir les acquittements  
25 Ack.1 à Ack.3, forment une file  $F_a$  stockée dans une mémoire tampon, en amont du point Pcd de contrôle de débit suivant le sens montant.

Dans le premier cas (figure 3) les segments de données des fenêtres de transmission de chaque connexion TCP se trouvent donc en grande partie stockés dans des files d'attente respectives, à l'interface avec le réseau ATM,  
30 au point de multiplexage des flots TCP dans le conduit virtuel, c'est à dire dans une mémoire du routeur assurant la l'interconnexion entre le réseau local et le réseau ATM. Elles peuvent alors occuper un espace mémoire allant jusqu'à

plusieurs dizaines de Kilo-octets par connexion (en fonction de la version de TCP mise en oeuvre) et par conséquent risquent de provoquer une congestion.

Dans l'état de la technique, on a déjà proposé un moyen d'éviter la constitution de ces files consistant à limiter la taille de la fenêtre Wu utilisable par l'émetteur en modifiant, dans les segments d'acquittement, la valeur du paramètre de taille de fenêtre Wa inséré par le récepteur. Ces techniques ont fait l'objet de différents travaux parmi lesquels on peut citer l'article de L. Kalampoukas, A. Varma, et K. K. Ramakrishnan, "Explicit Window Adaptation: A Method to Enhance TCP Performance," Proceedings of INFOCOM'98, avril 1998, qui, dans le cadre de la première solution décrite ci-dessus, propose un principe basé sur un algorithme de gestion de mémoire tampon permettant de déterminer directement la taille de fenêtre de la connexion TCP sans avoir à en connaître le débit. L'article de R. Satyavolu, K. Duvedi, et S. Kalyanaraman, "Explicit Rate Control of TCP Applications," ATM Forum/98-0152R1, février 1998, décrit un mécanisme de traduction d'une valeur de débit en une valeur de taille de fenêtre basé sur un des algorithmes d'allocation de débit développé pour la capacité de transfert ATM ABR. L'article de P. Narvaez et K.-Y. Siu, "An acknowledgement Bucket Scheme for Regulating TCP Flow over ATM", Computer Networks and ISDN Systems, vol. 30, Special Issue on ATM Traffic Management, 1998, propose l'implémentation d'un mécanisme plus complexe d'émulation de TCP au niveau du point de contrôle de débit associé à un "Ack bucket". Enfin, l'article de A. Koike, "TCP Flow Control with ACR Information," ATM Forum/97-0758, septembre 1997 suggère aussi l'utilisation d'un "Ack bucket". En outre, il est décrit dans la demande internationale de brevet WO 99/35790 un procédé et un dispositif d'optimisation du flux d'une connexion TCP à travers un réseau ATM à débit contrôlé avec une capacité de transfert de type ABR, consistant à contrôler le paramètre de taille de fenêtre présent dans les segments d'acquittement retournés par le récepteur en fonction du débit disponible dans le réseau ATM et de l'espace disponible dans la mémoire tampon stockant la file de la connexion. Néanmoins, ce procédé requiert la prise en compte d'une information relative au débit disponible dans la couche de protocole ABR/ATM.

L'invention vise à proposer une alternative aux méthodes connues de contrôle de flux d'une connexion TCP, notamment d'une connexion établie à travers un réseau à débit contrôlé, qui soit plus simple et donc plus facile à mettre en œuvre.

5 Ce but est atteint grâce à un procédé de contrôle du flux d'au moins une connexion TCP entre un émetteur et un récepteur, du type consistant à contrôler, au niveau d'un nœud de multiplexage déterminé par lequel transitent des segments TCP relevant de la connexion, un paramètre de taille de fenêtre présent dans des segments d'acquittement retournés par le récepteur,  
10 caractérisé en ce qu'il comprend les étapes consistant à :

a) recevoir au niveau dudit nœud de multiplexage déterminé un segment d'acquittement provenant du récepteur sur le sens montant (récepteur vers émetteur) de la connexion ;

b) contrôler un paramètre de taille de fenêtre présent dans ledit segment  
15 d'acquittement, en fonction de la différence entre, d'une part, une première valeur de contexte associée à la connexion TCP, définie comme le numéro de séquence du dernier segment ayant été transmis depuis ledit nœud de multiplexage déterminé sur le sens descendant (émetteur vers récepteur) de la connexion auquel on ajoute la longueur dudit segment, et, d'autre part, le  
20 numéro de séquence indiqué dans le segment d'acquittement ;

c) transmettre sur le sens montant de la connexion, depuis ledit nœud de multiplexage vers l'émetteur, le segment d'acquittement avec le paramètre de taille de fenêtre ainsi contrôlé.

En particulier, la connexion TCP est établie à travers un réseau 30 à  
25 débit contrôlé. Le nœud de multiplexage déterminé est alors de préférence un point de contrôle du débit du réseau à débit contrôlé. Le point de contrôle du débit coopérant avec une mémoire pouvant contenir une file d'attente associée à la connexion, par laquelle transitent des segments de données émis par l'émetteur sur le sens descendant de la connexion, l'étape c) peut consister à  
30 contrôler, en fonction de ladite différence, le paramètre de taille de fenêtre présent dans ledit segment d'acquittement de manière à maintenir le volume des données stockées dans la file d'attente inférieur à une seconde valeur de contexte associée à la connexion TCP. De préférence, le nœud de

multiplexage déterminé est situé le plus près possible de l'émetteur, par exemple à l'interface entre le réseau local auquel est relié l'émetteur et le réseau à débit contrôlé. C'est notamment le cas, dans un exemple, lorsque ce nœud de multiplexage déterminé est aussi un point de contrôle de débit du

5 réseau à débit contrôlé

Le mécanisme de contrôle de la taille des fenêtres de transmission selon l'invention permet d'empêcher l'apparition d'un phénomène de congestion au niveau du point de contrôle de débit d'un réseau à débit contrôlé en débit, en limitant la taille de fenêtre de chaque connexion TCP de telle manière que la

10 taille de la file d'attente d'une connexion se constituant au niveau du point de contrôle de débit ne dépasse pas une valeur limite déterminée, qui est égale à la seconde valeur de contexte. Toutefois, il autorise des variations du débit offert aux connexions TCP par le réseau à débit contrôlé. Selon un autre avantage, ce mécanisme est compatible avec les algorithmes de contrôle de

15 congestion mentionnés en introduction, qui agissent sur la taille de la fenêtre annoncée telle qu'elle est définie dans les segments d'acquittement lors de leur émission par le récepteur. Le mécanisme de contrôle de la taille de la fenêtre annoncée vient se superposer à ces mécanismes de contrôle de congestion connus.

20 L'invention propose également une unité de Contrôle de flux pour au moins une connexion TCP entre un émetteur et un récepteur, qui comprend :

- des moyens pour recevoir des segments de données TCP en provenance de l'émetteur et les transmettre vers le récepteur sur le sens descendant (émetteur vers récepteur) de la connexion TCP, et pour

25 déterminer, à partir de chaque segment de données ainsi transmis, une première quantité indicative du rang, au sein d'un flot de données transmis suivant le sens descendant de la connexion TCP, d'un premier élément de données à transmettre vers le récepteur dans un prochain segment de données ;

30 - des moyens pour recevoir des segments d'acquittement TCP en provenance du récepteur et les transmettre vers l'émetteur sur le sens montant (récepteur vers émetteur) de la connexion TCP, et pour extraire de chaque segment d'acquittement TCP reçu une seconde quantité indicative du rang, au

sein dudit flot de données, d'un prochain élément de données attendu par le récepteur ;

- des moyens de régulation pour contrôler, en fonction de la différence entre ladite première quantité et ladite seconde quantité, un paramètre de taille de fenêtre ( $W_a$ ) présent dans ledit segment d'acquittement TCP reçu avant de transmettre ledit segment d'acquittement TCP vers le récepteur.

En utilisant le vocabulaire propre à TCP, ladite première quantité correspond au numéro de séquence présent dans le segment de données auquel on ajoute la longueur (en octets) du champ de données de ce segment, et ladite seconde quantité correspond au numéro de séquence présent dans le segment d'acquittement TCP.

Dans un exemple, l'unité de contrôle de flux comprend en outre une mémoire pour contenir une file d'attente de segments de données reçus de l'émetteur et à transmettre vers le récepteur sur le sens descendant de la connexion TCP à travers un réseau à débit contrôlé, et en ce que le contrôle dudit paramètre de taille de fenêtre exercé par les moyens de régulation comporte une limitation dudit paramètre à une valeur au plus égale à la somme de ladite différence et d'une taille maximum fixée pour la file d'attente dans la mémoire. Cette taille maximum correspond à la seconde valeur de contexte associée à la connexion TCP, mentionnée plus haut.

L'invention propose encore un équipement d'interconnexion entre un premier réseau, par exemple un réseau local, et un second réseau, par exemple un réseau à débit contrôlé, qui comprend une telle unité de contrôle de flux.

D'autres caractéristiques et avantages de l'invention apparaîtront encore à la lecture de la description qui va suivre. Celle-ci est purement illustrative et doit être lue en regard des dessins annexés, sur lesquels on a représenté :

- à la figure 1, déjà analysée : un schéma illustrant le principe de fenêtre glissante du mécanisme de contrôle de flux de TCP ;
- à la figure 2, également déjà analysée : un schéma illustrant les transferts des segments et d'acquittements dans le réseau ;
- à la figure 3 et à la figure 4, également déjà analysées : un schéma illustrant le principe de remplissage d'une file de segments et d'une file

d'acquittements dans le cas, respectivement, d'un contrôle de débit d'une connexion TCP par espacement des segments ou par espacement des acquittements ;

5       - à la figure 5 : le schéma d'une connexion TCP adaptée à la mise en œuvre du procédé de l'invention ;

      - à la figure 6 : le format d'un datagramme IP encapsulant un segment TCP ;

      - à la figure 7 : le détail de l'en-tête d'un segment TCP ;

10      - à la figure 8 et à la figure 9 : des organigrammes montrant des étapes du procédé selon l'invention.

      A la figure 5, on a représenté schématiquement un exemple de liaison TCP sur un réseau à débit contrôlé adapté à la mise en œuvre du procédé selon l'invention. On considère un programme d'application s'exécutant sur un ordinateur 10 (ci-après l'émetteur) et devant transmettre des données à un  
15      programme d'application s'exécutant sur un ordinateur 20 distant (ci-après le récepteur). Ces deux programmes d'application utilisent le protocole TCP pour communiquer. L'émetteur 10 est relié à un réseau local 50, par exemple un réseau Ethernet à haut débit. Un tel réseau ne génère pas de congestion, compte tenu de l'importance du débit offert par rapport au volume des données  
20      qu'il transmet. Le récepteur 20 est également relié à un réseau local 60. Dans un exemple, on considère que le réseau local 60 est également un réseau ethernet, mais il peut bien entendu s'agir d'un réseau différent, par exemple de type LAN (de l'anglais "Local Area Network").

      La connexion TCP se fait sur un réseau à débit commandé 30, par  
25      exemple un réseau ATM avec une capacité de transfert de type ABR. Le sens descendant de la connexion est désigné par une simple flèche, alors que son sens montant est désigné par une double flèche. Un équipement 40 d'interconnexion entre le réseau local 50 et le réseau 30 est relié à l'entrée du réseau 30. Il s'agit d'un routeur de périphérie. Cet équipement est le siège du  
30      multiplexage des différentes connexions TCP établies sur le réseau 30. Dans un exemple, il est aussi le siège du contrôle de débit du réseau 30 dans le sens descendant et comporte à cet effet un point de contrôle de débit Pcd. Il est enfin le siège du contrôle de la taille de la fenêtre annoncée qui est à la base

du procédé de contrôle de flux de la connexion TCP selon l'invention. Un équipement 70 d'interconnexion entre le réseau 30 et le réseau 60 est disposé à la sortie du réseau 30 pour assurer la fonction inverse de démultiplexage des segments TCP. Il s'agit également d'un routeur de périphérie.

- 5            Pour l'établissement de la connexion l'émetteur 10 émet un segment avec un fanion SYN dans le sens descendant et le récepteur émet en retour un segment avec un fanion ACK dans le sens montant, précisant éventuellement, dans un champ d'option, la valeur MSS (de l'anglais "Maximum Segment Size") de la taille maximum des segments qu'il peut recevoir compte tenu des
- 10    caractéristiques de sa mémoire tampon. La connexion TCP est définie par la combinaison du numéro de port source et du numéro de port destination qui identifient respectivement le programme d'application de l'émetteur 10 et celui du récepteur 20.

Le flot de données à émettre est découpé par TCP et encapsulé dans

15    des segments TCP dont la taille a été déterminée lors de l'établissement de la connexion compte tenu éventuellement de la valeur MSS. Chaque segment TCP est lui-même encapsulé dans un datagramme IP, au sein du routeur 40, selon le format montré à la figure 5. Le segment TCP comprend un bloc de données TCP précédé d'un en-tête TCP ("TCP header" en anglais) codé sur

20    au moins 20 octets. Le datagramme IP comprend le segment TCP précédé d'un en-tête IP codé également sur 20 octets. La figure 6 montre le format de l'en-tête TCP. Celui-ci comprend :

- un champ contenant le numéro de port source et un autre pour le numéro de port destination, qui sont codés sur 16 bits chacun ;
- 25        - un champ contenant le numéro de séquence (dans le cas d'un segment de données) et un autre pour le numéro de séquence d'acquittement (dans le cas d'un segment d'acquittement), qui sont codés sur 32 bits chacun ;
- un champ pour indiquer la longueur de l'en-tête (qui est toujours un multiple de 32 bits), codée sur quatre bits. ; ce champ est nécessaire car l'en-
- 30    tête contient éventuellement un (ou plusieurs) champ(s) d'option, codé chacun sur 32 bits. Un tel champ d'option peut servir à indiquer la valeur du paramètre MSS, mentionné plus haut, s'agissant d'un segment ACK retourné par le récepteur lors de l'établissement de la connexion ;



- 6 bits réservés pour un usage ultérieur, et 6 bits désignant respectivement la valeur de fanions URG, ACK, PSH, RST, SYN et FIN. Ces fanions sont actifs à 1. Le fanion URG sert à indiquer que certains au moins des octets de données du segment doivent être remis en urgence. Un pointeur d'urgence, contenu dans un autre champ de l'en-tête, indique alors quel est le premier de ces octets qui doivent être remis en urgence. Les fanions ACK, RST, SYN, et FIN ont été décrits plus haut. Enfin, le fanion PSH indique que les données sont émises en utilisant une opération appelée "push" en anglais, exécuté un programme d'application pour obtenir l'émission immédiate des données ;

- un champ "taille de fenêtre", pour contenir un paramètre de taille de fenêtre  $W_a$ , codé sur 16 bits ; un tel paramètre  $W_a$  est présent dans chaque segment d'acquittement retourné par le récepteur sur le sens montant de la connexion ; il permet au récepteur d'indiquer à l'émetteur le nombre d'octets qu'il est prêt à accepter ;

- un champ "checksum TCP", indiquant le résultat du calcul d'un total de vérification, codé sur 16 bits, afin de permettre la détection d'erreurs de transmission ;

- un champ pour contenir le pointeur d'urgence précité, codé sur 16 bits ;  
- et enfin le ou les champs d'option précités.

Comme on l'aura compris, le format de l'en-tête est identique pour les segment émis par l'émetteur et les segments d'acquittements émis par le récepteur, la valeur de certains champs propres à l'un ou l'autre seulement de ces deux types de segment étant indifférente au programme d'application respectivement du récepteur et de l'émetteur.

Les étapes du procédé selon l'invention vont maintenant être décrites en référence à l'organigramme de la figure 8. La première étape 100 consiste à recevoir, au niveau du point de contrôle de débit  $P_{cd}$ , un segment d'acquittement, noté  $Ack_{i,j}$  dans la suite, qui est par exemple le  $j$ -ième acquittement émis par le récepteur depuis l'établissement de la connexion d'indice  $i$ , où  $i$  et  $j$  sont des indices entiers. Ce segment d'acquittement provient donc du récepteur dans le sens montant de la connexion d'indice  $i$ .

Dans une étape de 200, le segment d'acquittement  $Ack_{i,j}$  est décapsulé pour détecter la valeur du numéro de séquence  $NoSeqAck_{i,j}$  de  $Ack_{i,j}$ .

Parallèlement, dans une étape 300, un contexte de connexion  $\{NoSeqData_i, Lim\}$  est lu dans une mémoire de contexte sur laquelle on reviendra plus loin.

- 5 Ce contexte contient deux paramètres ou valeurs de contexte, qui sont associées à la connexion.

La première valeur de contexte  $NoSeqData_i$  est définie comme le numéro de séquence du dernier segment ayant transité par le point de contrôle de débit sur le sens descendant de la connexion auquel on ajoute la longueur  
10 de ce segment, exprimée en nombre d'octets. Par longueur d'un segment de données, on entend plus précisément la longueur d'un champ de données du segment, c'est à dire la longueur totale du segment moins celle de son en-tête. Cette première valeur de contexte correspond donc au rang, dans le flot de données transmis sur le sens descendant de la connexion TCP, d'un premier  
15 élément de données ou octet à transmettre vers le récepteur dans un prochain segment de données. On reviendra plus loin sur la manière dont un contexte est généré et sur celle dont la première valeur de contexte est mise à jour.

La seconde valeur de contexte  $Lim$  correspond selon l'invention à une taille maximale fixée pour la file d'attente  $F_s$  dans la mémoire tampon dans  
20 laquelle transitent les segments émis par l'émetteur sur le sens descendant de la connexion. Cette valeur est constante pendant toute la durée de la connexion. Elle est associée à la connexion. Dans un exemple, elle est déterminée à l'établissement de la connexion TCP. Dans un mode de réalisation, elle est la même pour toutes les connexions. Toutefois, dans un  
25 mode de réalisation préféré, elle est déterminée en fonction du paramètre de taille maximale de segment MSS associé à la connexion. De préférence, elle est supérieure à la valeur dudit paramètre de taille maximale de segment MSS, de manière à éviter le phénomène de fenêtre stupide ("Silly Window", en anglais). On rappelle que la valeur de ce paramètre est le cas échéant indiquée  
30 par le récepteur à l'ouverture de la connexion. La seconde valeur de contexte  $Lim$  peut aussi être générée dynamiquement en fonction de l'espace mémoire disponible pour les files d'attente des nouvelles connexions TCP.

Dans une étape 400, on calcule un paramètre de contrôle Diff défini comme la différence entre, d'une part, la première valeur de contexte  $NoSeqData_i$  associée à la connexion d'indice  $i$ , et, d'autre part, le numéro de séquence  $NoSeqAck_{i,j}$  indiqué dans le segment d'acquiescement  $Ack_{i,j}$ . Dit  
5 autrement  $Diff = NoSeqData_i - NoSeqAck_{i,j}$ . Les valeurs positives du paramètre Diff constituent une estimation par excès du volume Vdata des données en cours de transfert dans le réseau sur le sens descendant de la connexion d'indice  $i$ . Il s'agit d'une estimation par excès en ce sens que la valeur d'estimation produite correspond au volume maximum que le réseau  
10 peut être en train d'absorber sur le sens descendant au moment où cette étape est exécutée. En effet, certains segments peuvent avoir été remis au récepteur depuis le moment où celui-ci a émis le segment d'acquiescement  $Ack_{i,j}$ . D'autres segments peuvent avoir été détruits ou perdus. Bien entendu la paramètre Diff n'est représentatif du volume de données en cours de transfert dans le réseau  
15 sur le sens descendant de la connexion que si il est positif. On verra sur des exemples que tel n'est pas forcément le cas, en raison du mécanisme de retransmission de segment de TCP.

Une étape 500 consiste ensuite à calculer une valeur modifiée d'un paramètre de taille de fenêtre  $Wa$  inséré par le récepteur dans le segment  
20 d'acquiescement  $Ack_{i,j}$ . Cette étape permet de modifier dans ledit segment d'acquiescement, en fonction du paramètre Diff produit à l'étape précédente, la taille de la fenêtre annoncée de manière à maintenir la taille de la file d'attente  $Fs$  inférieure une taille maximale fixée, qui correspond à la seconde valeur de contexte  $Lim$  associée à la connexion d'indice  $i$ . On rappelle que la valeur de  
25 ce paramètre  $Wa$  est contenue dans le champ "taille de fenêtre" du segment TCP. Dans un exemple, le paramètre taille de fenêtre  $Wa$  est contrôlé selon la règle suivante :

soit  $Wa = \text{Min}(Diff + Lim, Wa)$  si  $Diff \geq 0$   
soit  $Wa = \text{Min}(Lim, Wa)$  si  $Diff < 0$

30 où Min est la fonction minimum.

De cette manière, la valeur du paramètre de taille de fenêtre  $W_a$  est en permanence limitée à une valeur au plus égale à la somme du paramètre de contrôle Diff et de la seconde valeur de contexte  $Lim$ .

Dans une étape 600, la valeur contenue dans le champ "checksum TCP" du segment d'acquittement  $Ack_{i,j}$  est recalculée et le champ "checksum TCP" est modifié en conséquence, de manière à tenir compte de la modification du champ "taille de fenêtre" contenant la valeur du paramètre de taille de fenêtre  $W_a$ . Dans une ultime étape 700, le segment d'acquittement  $Ack_{i,j}$ , avec un champ "taille de fenêtre" et un champ "Checksum TCP" éventuellement modifiés selon l'invention, est transmis, de préférence sans délai, vers l'émetteur 10 pour lui être remis.

Les étapes 100 à 700 décrites ci-dessus sont exécutées pour chaque segment d'acquittement émis par le récepteur 20 et transitant par le point de contrôle de débit Pcd sur le sens montant de la connexion TCP..

15 L'étape de calcul 400 est mise en oeuvre au niveau du point de contrôle du débit Pcd du réseau à débit contrôlé 30, c'est à dire aussi au niveau de la file d'attente  $F_s$ . De plus, le point de contrôle du débit Pcd est préférentiellement le plus près possible de l'émetteur 10. De cette manière, la portion du réseau située entre l'émetteur et le point de contrôle de débit Pcd  
20 contient le moins de données possible. De la sorte, le paramètre Diff est une estimation la plus précise possible du volume de donnée  $V_{data}$  en cours de transfert dans le réseau sur le sens descendant de la connexion. Dans un exemple préféré, le point de contrôle du débit Pcd est situé dans l'équipement 40 d'interconnexion entre le réseau local 50 auquel est relié l'émetteur 10 et le  
25 réseau à débit contrôlé 30.

L'équipement d'interconnexion entre le réseau local auquel est relié l'émetteur et le réseau à débit contrôlé, tel que le routeur 40 de la figure 5 comprend des moyens pour la mise en oeuvre du procédé. En particulier, il comprend le point de contrôle du débit Pcd et une mémoire de contexte MC. La  
30 seconde valeur de contexte  $Lim$  et/ou la première valeur de contexte  $NoSeqData_i$  sont lues dans cette mémoire MC à l'étape 300 de la figure 8. La première valeur de contexte associée à une connexion déterminée établie sur le réseau 30 est mise à jour dans la mémoire MC à chaque passage d'un

segment au niveau du point de contrôle du débit Pcd sur le sens descendant de cette connexion. A la figure 9, on a représenté un organigramme des étapes de cette mise à jour. Dans une étape 1000, un segment de données TCP, noté  $Seg_{i,k}$ , qui est par exemple le k-ième segment de données émis par l'émetteur

5 10 sur le sens descendant de la connexion d'indice i, où k est un autre indice entier, est reçu au niveau du point de contrôle de débit Pcd. Dans une étape 2000, la première valeur de contexte *NoSeqData<sub>i</sub>* associée à la connexion d'indice i est mise à jour dans la mémoire de contexte MC. A cet effet, une copie du segment  $Seg_{i,k}$  est décapsulée pour extraire son numéro de

10 séquence. La longueur du segment est alors déterminée et ajoutée à son numéro de séquence et la valeur obtenue est enregistrée en tant que mise à jour de la première valeur de contexte *NoSeqData<sub>i</sub>*. La longueur du segment, exprimée en nombre d'octets, est obtenue en soustrayant la longueur de l'en-tête TCP (indiquée dans un champ de l'en-tête du segment TCP comme il a

15 été exposé ci-dessus en regard de la figure 7) à la longueur de la charge utile du datagramme IP. Cette dernière longueur est calculée à partir de la longueur totale du datagramme IP à laquelle est soustraite la longueur de son en-tête (indiquée quant à elle dans l'en-tête du datagramme IP). Enfin, dans une étape 3000, le segment de données  $Seg_{i,k}$  est émis (sans modification) sur le sens

20 descendant de la connexion vers le récepteur 20.

On notera que la mémoire de contexte MC stocke avantageusement les contextes de chacune des connexions TCP établies sur le réseau à débit contrôlé 30. A cet effet, la mémoire MC est indexée par le couple formé du

25 numéro de port source et du numéro de port destination définissant la connexion. La capacité mémoire de la mémoire de contexte est gérée dynamiquement. En effet, un contexte de contrôle est créé lors de la détection d'une nouvelle connexion IP. Une nouvelle connexion TCP est détectée soit de manière explicite, à la suite de la détection de segments d'établissements de

30 connexion (segments portant le fanion SYN ainsi qu'il a été dit en introduction), soit de manière implicite à la suite de la détection de segments appartenant à une connexion pour laquelle aucun contexte n'existe encore. Pour permettre ces détections et ces créations de nouveaux contextes, une copie des

segments émis par l'émetteur est systématiquement décapsulée pour en déduire, notamment, le couple formé du numéro de port source et du numéro de port destination, qui identifient la connexion concernée. De la même manière, un contexte est libéré de la mémoire, soit de manière explicite, à la  
5 suite de la détection de segments de fermeture de connexion (segments portant le fanion FIN), soit de manière implicite à la suite de la détection de l'expiration d'une temporisation ad-hoc. Bien entendu, l'espace mémoire occupé par la mémoire de contexte selon l'invention est bien plus petit que celui occupé par une file d'attente  $F_s$  selon l'art antérieur, en sorte que  
10 l'invention permet une économie substantielle d'espace mémoire.

A titre d'exemple du déroulement du procédé de l'invention, on considère un flot de données partagé en segments de taille constante égale à 100 octets, à transmettre sur une connexion TCP pour laquelle le premier contexte  $Lim$  a la valeur 300 ( $Lim=300$ ). Dit autrement, on cherche à limiter la  
15 taille de la file d'attente  $F_s$  de cette connexion à 300 octets, c'est à dire l'équivalent de 3 segments.

Si l'on se place à un moment où le dernier segment émis par l'émetteur contenait les octets 401 à 500 du flot de données, la première valeur de contexte  $NoSeqData$  est affectée de la valeur  $401+100=501$ . Si la taille  
20 courante de la fenêtre de transmission est égale à 400 ( $W=400$ ), cela signifie que, du point de vue de l'émetteur, le premier segment, qui portait les octets 1 à 100, a été valablement reçu et acquitté, alors que les deuxième, troisième, quatrième et cinquième segments, portant les octets respectivement 101 à, 200, 201 à 300, 301 à 400, et 401 à 500, et totalisant 400 octets à eux quatre,  
25 ont été émis mais n'ont pas encore été acquittés. Selon l'invention, on considère qu'ils sont en cours de transfert sur le sens descendant de la connexion TCP.

Admettons, dans un premier cas, que le récepteur émet un acquittement avec un numéro de séquence  $NoSeqAck$  égal à 201 et avec une taille de  
30 fenêtre annoncée égale à 500. Dit autrement, dans ce premier cas, le récepteur a reçu et acquitte le deuxième segment, et autorise un élargissement de la fenêtre  $W$  de 400 à 500 octets. On calcule alors le paramètre Diff par la différence :

$$\text{Diff} = \text{NoSeqAck} - \text{NoSeqData} = 501 - 201 = 300 ;$$

Comme cette valeur est positive, la valeur modifiée de la taille de fenêtre annoncée par le récepteur est donnée par :

$$W_a = \text{Min} (\text{Lim} + \text{Diff}, W_a) = \text{Min} (600, 500) = 500$$

5 Dit autrement, la taille de fenêtre  $W_a$  annoncée dans l'acquittement est maintenue. En d'autres termes, l'élargissement de la fenêtre de transmission  $W$  de la connexion est accepté. En effet, le procédé selon l'invention consiste à considérer que le sens descendant de la connexion peut au mieux absorber 600 octets puisque le réseau absorbe alors les 300 octets des troisième, 10 quatrième et cinquième segments qui sont considérés comme étant toujours en cours de transfert sur le sens descendant, et que le sens descendant dispose d'une capacité supplémentaire de 300 octets permise par la file d'attente  $F_s$ .

Admettons, dans un deuxième cas, que le récepteur émet un acquittement avec un numéro de séquence  $\text{NoSeqAck}$  égal à 401 et avec une 15 taille de fenêtre annoncée égale à 500 ( $W_a = 500$ ). Dit autrement, dans ce deuxième cas, le récepteur a reçu et acquitte le deuxième, le troisième, et le quatrième segment, et autorise un élargissement de la fenêtre  $W$  de 400 à 500 octets. On calcule alors le paramètre  $\text{Diff}$  par la différence :

$$\text{Diff} = \text{NoSeqAck} - \text{NoSeqData} = 501 - 401 = 100 ;$$

20 Comme cette valeur est positive, la valeur modifiée de la taille de fenêtre annoncée par le récepteur est donnée par :

$$W_a = \text{Min} (\text{Lim} + \text{Diff}, W_a) = \text{Min} (400, 500) = 400$$

Dit autrement, la taille de fenêtre  $W_a$  annoncée dans l'acquittement est modifiée pour être ramenée de 500 à 400 octets. En d'autres termes, 25 l'élargissement de la fenêtre de transmission  $W$  de la connexion est refusé. En effet, le procédé selon l'invention consiste à considérer que le sens descendant de la connexion ne peut au mieux absorber que 400 octets, puisque le réseau n'absorbe à ce moment que les 100 octets du cinquième segment qui sont dorénavant les seuls (au plus) en cours de transfert, et que le sens descendant 30 ne bénéficie que d'une capacité supplémentaire de 300 octets permise par la file d'attente  $F_s$ . Permettre l'émission de quatre nouveaux segments risquerait donc de provoquer une congestion au niveau de la file d'attente  $F_s$  entraînant

un dépassement de la valeur maximum de la capacité de celle-ci telle qu'elle est définie par la seconde valeur de contexte *LIM*.

Si l'on se place maintenant à un moment où, ayant transmis les quatre premiers segments, dont seul le premier a valablement été acquitté, mais  
5 n'ayant pas reçu l'acquiescement correspondant au deuxième segment (octets 101 à 200) avant l'expiration de la temporisation associée à celui-ci, l'émetteur vient de le retransmettre. La première valeur de contexte *NoSeqData* est donc affectée de la valeur  $101+100=201$ . Les deuxième, troisième, quatrième et  
10 cinquième segments, portant les octets respectivement 101 à 200, 201 à 300, 301 à 400, et 101 à 200, et totalisant 400 octets à eux quatre, ont été émis mais n'ont pas encore été acquittés. Admettons que le récepteur émet alors un acquiescement avec un numéro de séquence *NoSeqAck* égal à 401 et avec une  
15 taille de fenêtre annoncée égale à 500. Dit autrement, dans ce premier cas, le récepteur a reçu et acquitte le deuxième, le troisième, et le quatrième segment, et autorise un élargissement de la fenêtre *W* de 400 à 500 octets. On calcule alors le paramètre *Diff* par la différence :

$$\text{Diff} = \text{NoSeqAck} - \text{NoSeqData} = 201 - 401 = -200 ;$$

Comme cette valeur est négative, elle ne peut correspondre à un volume de données transitant sur le sens descendant de la connexion. La valeur  
20 modifiée de la taille de fenêtre annoncée par le récepteur est donnée par :

$$W_a = \text{Min}(\text{Lim}, W_a) = \text{Min}(300, 500) = 300$$

Dit autrement, la taille de fenêtre *W<sub>a</sub>* annoncée dans l'acquiescement est modifiée pour être ramenée de 500 à 300 octets. En d'autres termes, l'élargissement de la fenêtre de transmission *W* de la connexion est refusé, et  
25 au contraire, la taille de la fenêtre *W* est ramenée de 400 à 300 du fait de la nouvelle valeur de la taille de la fenêtre annoncée *W<sub>a</sub>* obtenue selon l'invention. En effet, le procédé selon l'invention consiste à considérer que le réseau ne peut plus absorber de données, que seuls les 300 octets de la file d'attente *F<sub>s</sub>* sont disponibles pour absorber les données de la fenêtre *W*, et  
30 qu'il convient donc de limiter la taille de la fenêtre *W* à la taille limite *Lim* de la file *F<sub>s</sub>*.

On notera que le mécanisme de contrôle de la taille de la fenêtre annoncée conforme au procédé selon l'invention est avantageux car il ne



requiert pas d'indication explicite sur le débit de la connexion. De plus il est compatible avec les mécanismes de contrôle de la taille de la fenêtre mis en œuvre au niveau du récepteur, tels que les algorithmes de démarrage lent, d'évitement de congestion, de retransmission rapide et de recouvrement

5 rapide.

## REVENDEICATIONS

1. Procédé de contrôle du flux d'au moins une connexion TCP entre un  
5 émetteur (10) et un récepteur (20), du type consistant à contrôler, au niveau  
d'un nœud de multiplexage déterminé par lequel transitent des segments TCP  
relevant de la connexion, un paramètre de taille de fenêtre ( $W_a$ ) présent dans  
des segments d'acquittement retournés par le récepteur, caractérisé en ce qu'il  
comprend les étapes consistant à :

10 a) recevoir (100) au niveau dudit nœud de multiplexage déterminé un  
segment d'acquittement provenant du récepteur sur le sens montant (récepteur  
vers émetteur) de la connexion ;

b) contrôler (200-600) un paramètre de taille de fenêtre ( $W_a$ ) présent  
dans ledit segment d'acquittement, en fonction de la différence (Diff) entre,  
15 d'une part, une première valeur de contexte ( $NoSeqData_i$ ) associée à la  
connexion TCP, définie comme le numéro de séquence du dernier segment  
ayant été transmis depuis ledit nœud de multiplexage déterminé sur le sens  
descendant (émetteur vers récepteur) de la connexion auquel on ajoute la  
longueur dudit segment, et, d'autre part, le numéro de séquence indiqué dans  
20 ledit segment d'acquittement ;

c) transmettre (700) sur le sens montant de la connexion, depuis ledit  
nœud de multiplexage vers l'émetteur (10), le segment d'acquittement avec un  
paramètre de taille de fenêtre ( $W_a$ ) ainsi contrôlé.

2. Procédé selon la revendication 1, caractérisé en ce que la connexion  
25 TCP est établie à travers un réseau (30) à débit contrôlé.

3. Procédé selon la revendication 2, caractérisé en ce que ledit nœud de  
multiplexage déterminé est un point de contrôle du débit (Pcd) du réseau à  
débit contrôlé (30).

4. Procédé selon la revendication 3, caractérisé en ce que, le point de  
30 contrôle du débit (Pcd) coopérant avec une mémoire pour contenir une file  
d'attente (Fs) associée à la connexion, par laquelle transitent des segments de  
données émis par l'émetteur sur le sens descendant de la connexion, l'étape c)  
consiste à contrôler, en fonction de ladite différence (Diff), le paramètre de taille  
de fenêtre ( $W_a$ ) présent dans ledit segment d'acquittement de manière à

maintenir le volume des données stockées dans la file d'attente ( $F_s$ ) inférieur à une seconde valeur de contexte ( $Lim$ ) associée à la connexion TCP.

5 Procédé selon l'une des revendications précédentes, caractérisé en ce que ledit nœud de multiplexage déterminé est situé le plus près possible de l'émetteur (10).

6. Procédé selon l'une des revendications 2 à 4, caractérisé en ce que ledit nœud de multiplexage déterminé est situé à une interface du réseau à débit contrôlé (30).

10 7. Procédé selon la revendication 4, caractérisé en ce que l'étape b) comprend d'une part la production (200-400) d'un paramètre de contrôle ( $Diff$ ) égal à ladite différence entre la première valeur de contexte ( $NoSeqData_i$ ) associée à la connexion TCP et le numéro de séquence indiqué dans le segment d'acquittement, et d'autre part le contrôle (500-600) du paramètre de taille de fenêtre ( $Wa$ ) présent dans le segment d'acquittement selon la règle :

15 soit  $Wa = \text{Min}(Diff + Lim, Wa)$  si  $Diff \geq 0$

soit  $Wa = \text{Min}(Lim, Wa)$  si  $Diff < 0$

où  $Wa$  est ledit paramètre de taille de fenêtre,

$Diff$  est ledit paramètre de contrôle,

$\text{Min}$  est la fonction minimum,

20 et  $Lim$  est ladite seconde valeur de contexte associée à la connexion.

8. Procédé selon l'une des revendications précédentes, caractérisé en ce que, à l'étape b), la seconde valeur de contexte et/ou la première valeur de contexte sont lues dans une mémoire de contexte (MC).

25 9. Procédé selon la revendication 8, caractérisé en ce que, la mémoire de contexte (MC) stocke les première et seconde valeurs de contexte de chacune des connexions TCP établies à travers le réseau à débit contrôlé (30).

30 10. Procédé selon l'une des revendications 8 ou 9, caractérisé en ce que la première valeur de contexte associée à une connexion déterminée est mise à jour dans la mémoire de contexte à chaque réception, au niveau dudit nœud déterminé, d'un segment de données provenant de l'émetteur sur le sens descendant de cette connexion.

11. Procédé selon l'une quelconque des revendications 4 à 10, caractérisé en ce que la seconde valeur de contexte (*Lim*) associée à la connexion TCP, est déterminée à l'établissement de la connexion TCP.

5 12. Procédé selon la revendication 11, caractérisé en ce que la seconde valeur de contexte (*Lim*) associée à la connexion TCP, est déterminée en fonction d'un paramètre de taille maximale de segment (MSS) associé à cette connexion.

10 13. Procédé selon la revendication 12, caractérisé en ce que la seconde valeur de contexte (*Lim*) associée à la connexion TCP est supérieure à la valeur dudit paramètre de taille maximale de segment (MSS) associé à cette connexion.

14. Procédé selon l'une des revendications 3 à 13, caractérisé en ce que le point de contrôle de débit (Pcd) est situé le plus près possible de l'émetteur (10).

15 15. Procédé selon l'une des revendications précédentes caractérisé en ce que les étapes a) à c) sont mises en œuvre dans un équipement d'interconnexion entre un réseau local (50) et un réseau à débit contrôlé (30) à travers lequel est établie la connexion TCP.

20 16. Procédé selon l'une des revendications précédentes caractérisé en ce que les étapes a) à c) sont exécutées pour chaque segment d'acquittement reçu par ledit nœud déterminé.

17. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que, à l'étape c), le segment d'acquittement est transmis sans délai.

25 18. Unité de Contrôle de flux pour au moins une connexion TCP entre un émetteur (10) et un récepteur (20), caractérisée en ce qu'elle comprend :

30 - des moyens pour recevoir des segments de données TCP en provenance de l'émetteur (10) et les transmettre vers le récepteur (20) sur le sens descendant (émetteur vers récepteur) de la connexion TCP, et pour déterminer, à partir de chaque segment de données ainsi transmis, une première quantité indicative du rang, au sein d'un flot de données transmis suivant le sens descendant de la connexion TCP, d'un premier élément de

données à transmettre vers le récepteur (20) dans un prochain segment de données ;

5 - des moyens pour recevoir des segments d'acquittement TCP en provenance du récepteur (20) et les transmettre vers l'émetteur (10) sur le sens montant (récepteur vers émetteur) de la connexion TCP, et pour extraire de chaque segment d'acquittement TCP reçu une seconde quantité indicative du rang, au sein dudit flot de données, d'un prochain élément de données attendu par le récepteur (20) ; et

10 - des moyens de régulation pour contrôler, en fonction de la différence entre ladite première quantité et ladite seconde quantité, un paramètre de taille de fenêtre (W<sub>a</sub>) présent dans ledit segment d'acquittement TCP reçu avant de transmettre ledit segment d'acquittement TCP vers l'émetteur (10).

15 19. Unité de contrôle de flux selon la revendication 18, caractérisée en ce qu'elle comprend en outre une mémoire pour contenir une file d'attente de segments de données reçus de l'émetteur (10) et à transmettre vers le récepteur (20) sur le sens descendant de la connexion TCP à travers un réseau (30) à débit contrôlé, et en ce que le contrôle dudit paramètre de taille de fenêtre (W<sub>a</sub>) exercé par les moyens de régulation comporte une limitation dudit paramètre à une valeur au plus égale à la somme de ladite différence et

20 d'une taille maximum fixée pour la file d'attente dans la mémoire.

20. Equipement d'interconnexion (40) entre un premier réseau (50) et un second réseau (30), caractérisé en ce qu'il comprend une unité de contrôle de flux selon l'une des revendications 18 ou 20.

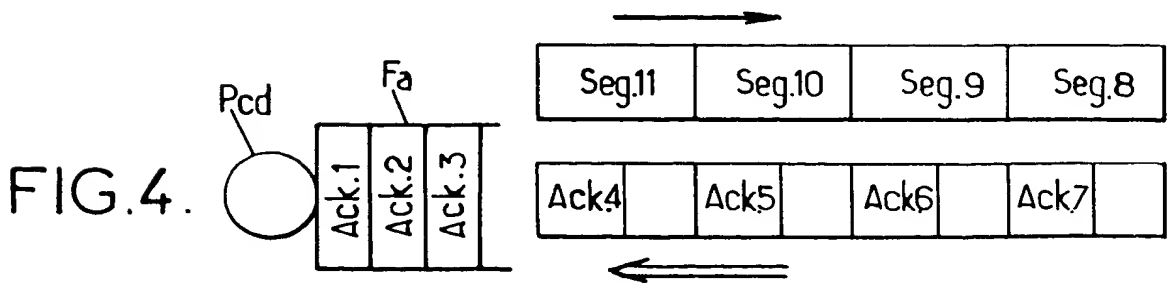
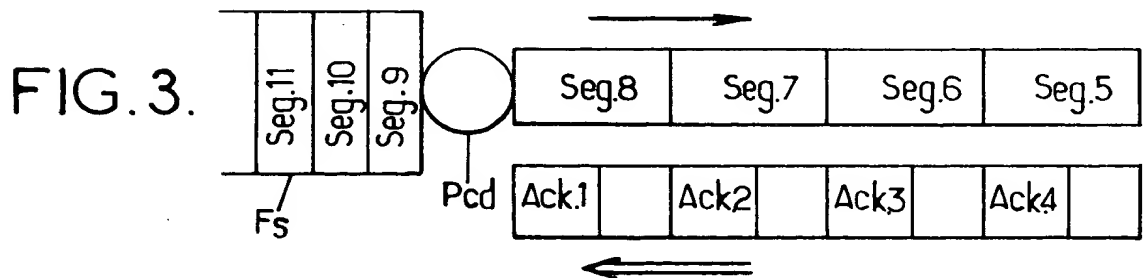
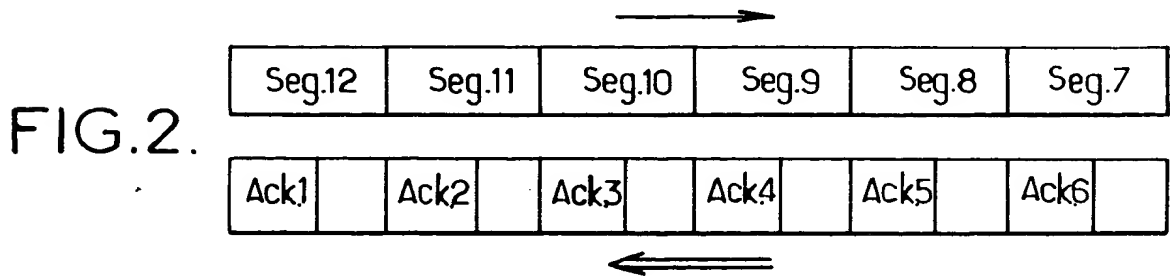
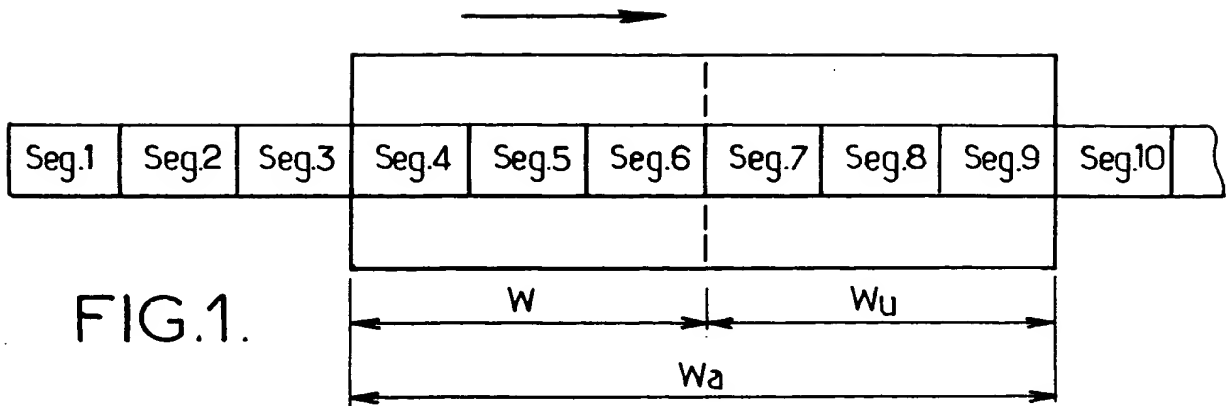
données à transmettre vers le récepteur (20) dans un prochain segment de données ;

5 - des moyens pour recevoir des segments d'acquittement TCP en provenance du récepteur (20) et les transmettre vers l'émetteur (10) sur le sens montant (récepteur vers émetteur) de la connexion TCP, et pour extraire de chaque segment d'acquittement TCP reçu une seconde quantité indicative du rang, au sein dudit flot de données, d'un prochain élément de données attendu par le récepteur (20) ; et

10 - des moyens de régulation pour contrôler, en fonction de la différence entre ladite première quantité et ladite seconde quantité, un paramètre de taille de fenêtre ( $W_a$ ) présent dans ledit segment d'acquittement TCP reçu avant de transmettre ledit segment d'acquittement TCP vers l'émetteur (10).

15 19. Unité de contrôle de flux selon la revendication 18, caractérisée en ce qu'elle comprend en outre une mémoire pour contenir une file d'attente de segments de données reçus de l'émetteur (10) et à transmettre vers le récepteur (20) sur le sens descendant de la connexion TCP à travers un réseau (30) à débit contrôlé, et en ce que le contrôle dudit paramètre de taille de fenêtre ( $W_a$ ) exercé par les moyens de régulation comporte une limitation dudit paramètre à une valeur au plus égale à la somme de ladite différence et  
20 d'une taille maximum fixée pour la file d'attente dans la mémoire.

20. Equipement d'interconnexion (40) entre un premier réseau (50) et un second réseau (30), caractérisé en ce qu'il comprend une unité de contrôle de flux selon l'une des revendications 18 ou 19.



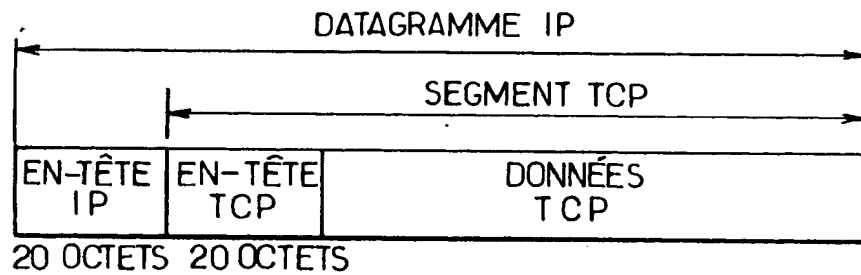
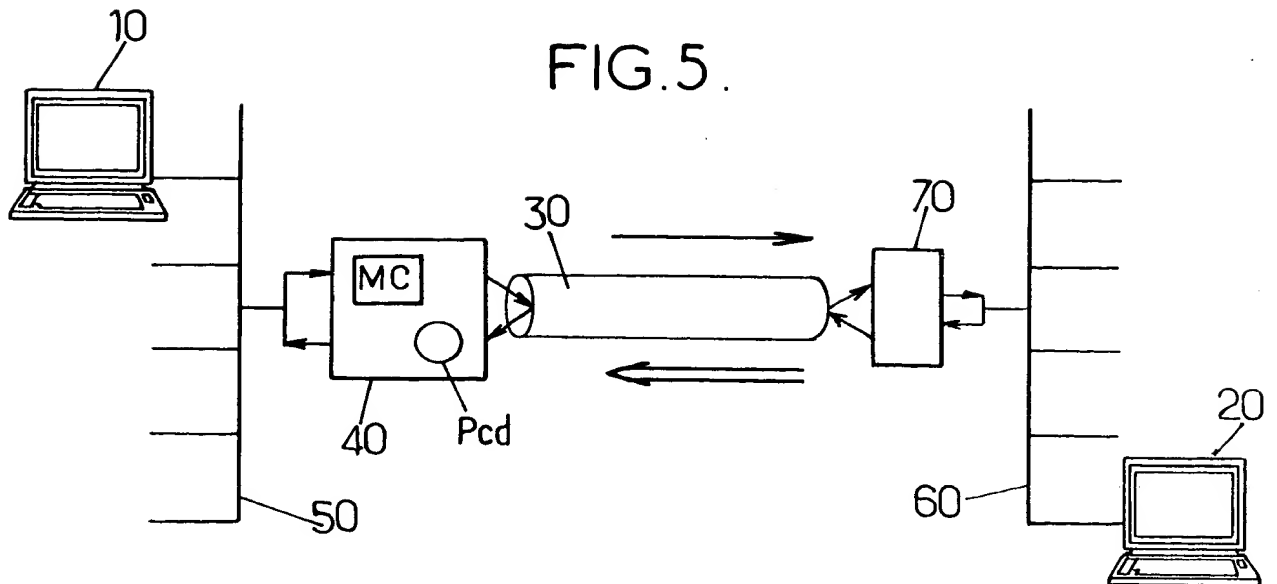


FIG.6.

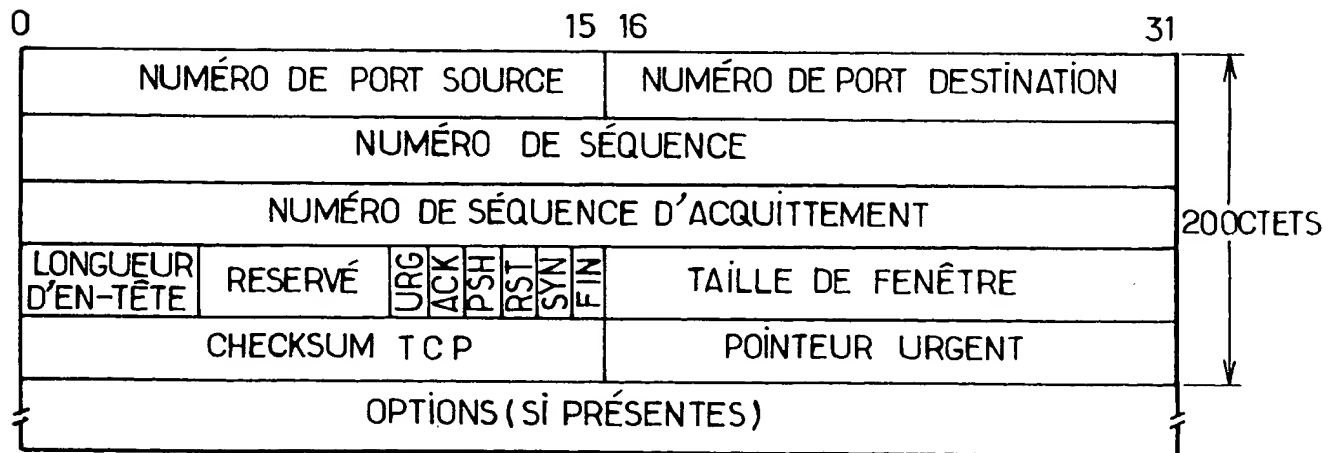




FIG.8.

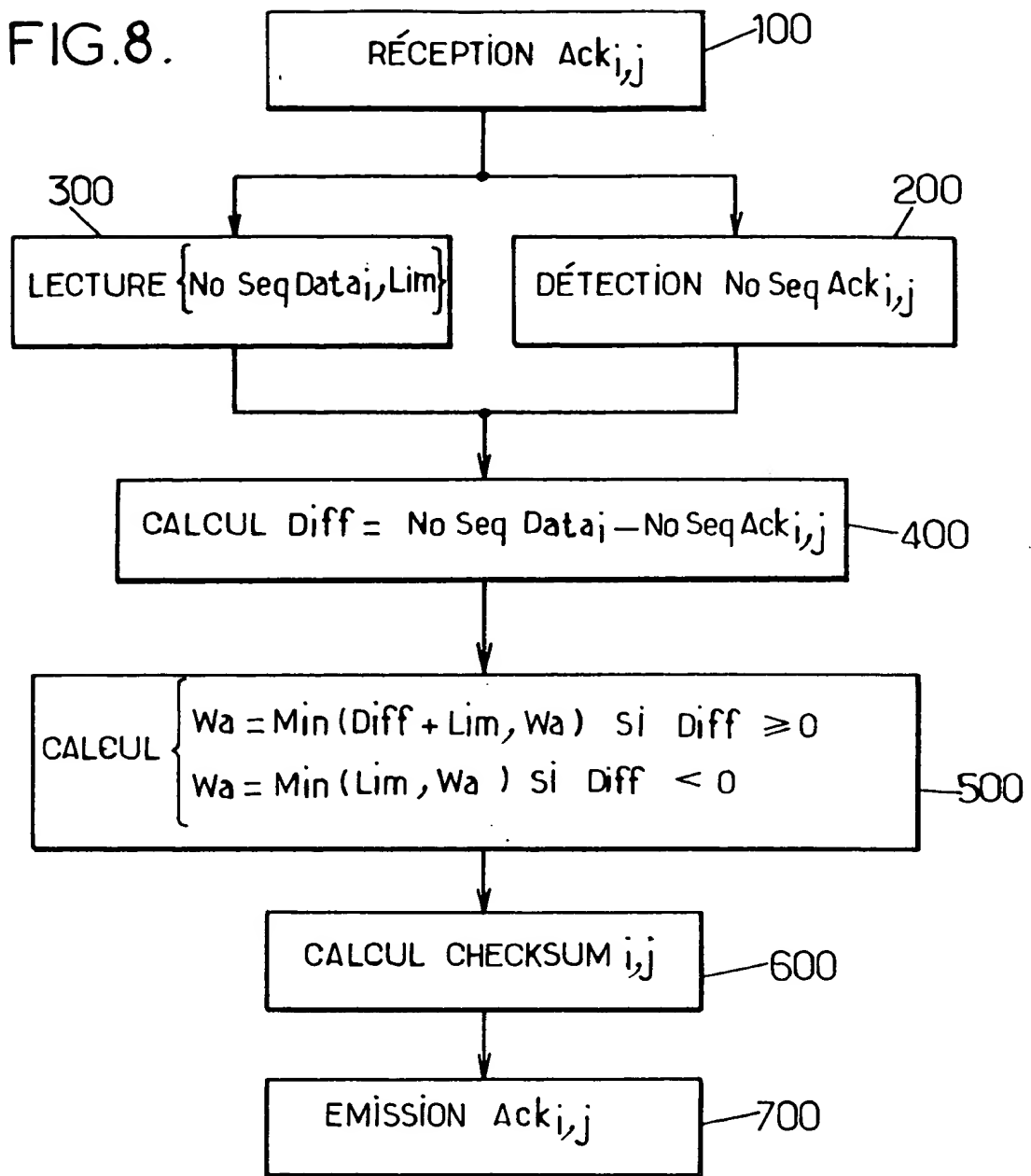
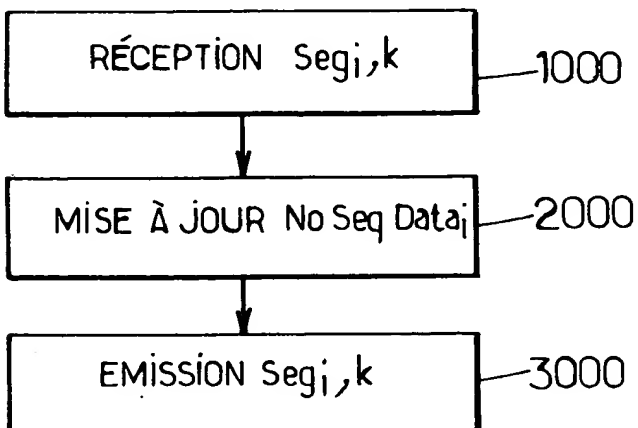


FIG.9.





22850

SERIAL NO.: 09/770,215  
FILING DATE: JAN. 29, 2001